

1                   1.     **(Amended)** A method of converting a source code containing a  
2     plurality of instructions in a predetermined order, including new instructions, to object code  
3     for use by a processor, the method including the steps of:  
4                   copying plurality of instructions to a temporary file, the new instructions each  
5     being copied as data in the form of object code corresponding to such instruction; and  
6                   applying the plurality of instructions of the temporary file to an assembler to  
7     produce object code corresponding to the old instructions and the data forming object code  
8     for the new instructions.

1                   2.     **(Amended)** A method of assembling source code containing existing  
2     machine language instructions and new machine language instructions with an existing  
3     assembler to produce object code having machine language instructions corresponding to  
4     each the instructions of the existing instruction set and the new instructions, the method  
5     including the steps of:  
6                   copying each of the existing machine language instructions to a temporary  
7     file;  
8                   copying each new machine language instruction to the temporary file as a  
9     data directive having a form corresponding to object code corresponding to such new  
10    machine language instruction; and  
11                  assembling the machine language instructions and the data directives to  
12    produce the object code.

Please add new claims 3-8

1                   3.     **(New)** A method of compiling source code having a plurality of first  
2     instructions and a at least one second instruction with a compiler capable of deciphering the first  
3     instructions but not the second instruction, including the steps of:  
4                   copying each of the first instructions to a temporary file;  
5                   converting the second instruction to an object code equivalent that forms an argument  
6     of a predetermined compiler statement that is written to the temporary file in place of the second  
7     instruction;  
8                   applying the temporary file to the compiler to

9 convert each of the first instructions to object code equivalents that are written  
10 to an object file; and  
11 removing the argument of the predetermined statement to write the argument to the  
12 object file.

A7 1 4. (New) The method of claim 3, wherein the first and second instructions are in  
2 a predetermined order in the source code, and the predetermined order is maintained when the first  
3 instructions and the include statement corresponding to the second statement are in the temporary  
4 file.

1 5. (New) The method of claim 3, wherein the predetermined statement is a data  
2 directive.

1 6. (New) A processing system operable to compile a source code having a  
2 plurality of first instructions and at least one second instruction to produce machine-readable code by  
3 copying each of the first instructions to a temporary file; and then,  
4 converting the second instruction to an object code equivalent that forms an argument  
5 of a predetermined compiler statement that is written to the temporary file in place of the second  
6 instruction;  
7 applying the temporary file to the compiler to  
8 convert each of the first instructions to object code equivalents that are written  
9 to an object file; and  
10 removing the argument of the predetermined statement to write the argument to the  
11 object file.

1 7. (New) The processing system of claim 6, wherein the predetermined compiler  
2 statement is a data directive statement.

1 8. (New) A system for source code compilation to produce machine-readable  
2 object code, the source code including a plurality of first instructions and at least one second  
3 instruction, the system including:  
4 a preprocessor operating to copy each of the first instructions to a temporary file, the  
5 second instruction first being converted to an object code equivalent and placed in an argument of a